Amendments to the Claims

Applicants submit no amendments to the claims. The following list of the currently pending claims is provided for the convenience of the Examiner.

Claim 1 (previously presented): In a computer system including a processor having a plurality of registers, a method for generating an aligned vector of first width from two second width vectors for single instruction multiple data (SIMD) processing, comprising the steps of:

loading a first vector from a memory unit into a first register, wherein the first vector contains a first byte of the aligned vector to be generated;

loading a second vector from the memory unit into a second register;

determining a starting byte in the first register wherein the starting byte specifies the first byte of the aligned vector;

extracting the aligned vector from the first register and the second register beginning from the first bit in the starting byte of the first register continuing through bits in the second register; and

replicating the aligned vector into a third register such that the third register contains a plurality of elements aligned for SIMD processing.

Claim 49 (previously presented): A method for generating an aligned vector from two source vectors for single instruction multiple data (SIMD) processing, comprising the steps of:

(1) loading a first source vector into a first register;

(2) loading a second source vector into a second register;

(3) reading a first plurality of elements from said first register and a second plurality of elements from said second register; and

(4) writing said first plurality of elements and said second plurality of elements into a third register in a particular order to produce a target vector having a plurality of elements aligned for SIMD processing.

Claim 50 (previously presented): The method as recited in claim 49, wherein said writing step comprises:

writing even-numbered, lower elements of said first register to said third register; and

writing sign bits of odd-numbered, lower elements of said first register to said third register.

Claim 51 (previously presented): The method as recited in claim 49, wherein said writing step comprises:

writing even-numbered, upper elements of said first register to said third register; and

writing sign bits of odd-numbered, upper elements of said first register to said third register.

Claim 52 (previously presented): A method for generating an ordered set of elements in a target vector from elements in a first source vector and a second source vector for single instruction multiple data (SIMD) vector processing, comprising the steps of:

(1) loading the first source vector into a first register;

(2) loading the second source vector into a second register;

(3) selecting a first subset of elements from said first register, said first subset comprising any one of the following groups of elements from the first source vector: odd elements, even elements, lower elements and upper elements; and

(4) selecting a second subset of elements from said second register, said second subset comprising any one of the following groups of elements from the second source vector: odd elements, even elements, lower elements and upper elements.

Claim 53 (previously presented): The method of claim 52, further comprising the step of:

(5) writing said first and said second subset of elements into a third register to facilitate a particular SIMD vector processing operation, said first subset being written into any one of the following groups of elements in said third register: upper elements, odd elements, and odd elements in reverse order, and said second subset being written into any one of the following groups of elements in said third register: lower elements, even elements, and even elements in reverse order, wherein elements written into said third register comprise the target vector.

Claim 54 (previously presented): The method as recited in claim 1, wherein the first width and second width are each 64 bits.

Claim 55 (previously presented): The method as recited in claim 54, wherein the third register is comprised of eight 8-bit elements.

Claim 56 (previously presented): The method as recited in claim 54, wherein the third register is comprised of four 16-bit elements.

Claim 57 (previously presented): The method as recited in claim 1, wherein the starting byte is specified as a variable in a register in an alignment instruction.

Claim 58 (previously presented): The method as recited in claim 1, wherein the first vector and the second vector are in contiguous locations in the memory unit.

Claim 59 (previously presented): The method as recited in claim 1, wherein the processor operates in a big-endian byte ordering mode.

Claim 60 (previously presented): The method as recited in claim 1, wherein the processor operates in a little-endian byte ordering mode.

Claim 61 (previously presented): The method as recited in claim 1, wherein the first vector and the second vector are each composed of eight 8-bit elements indexed from 0 to 7, and wherein said extracting step comprises:

extracting elements 4, 5, 6, and 7 from the first register.

Claim 62 (previously presented): The method as recited in claim 61, wherein said replicating step comprises:

replicating an element 0 of the third register from an element 4 of the first register;

replicating, for all bits of an element 1 of the third register, a sign bit of the element 4 of the first register;

replicating an element 2 of the third register from an element 5 of the first register;

replicating, for all bits of an element 3 of the third register, a sign bit of the element 5 of the first register;

replicating an element 4 of the third register from an element 6 of the first register;

replicating, for all bits of an element 5 of the third register, a sign bit of the element 6 of the first register;

replicating an element 6 of the third register from an element 7 of the first register; and

replicating, for all bits of an element 7 of the third register, a sign bit of the element 7 of the first register.

Claim 63 (previously presented): The method as recited in claim 1, wherein the first vector and the second vector are each composed of eight 8-bit elements indexed from 0 to 7, and wherein said extracting step comprises:

extracting elements 0, 1, 2 and 3 from the first register.

Claim 64 (previously presented): The method as recited in claim 63, wherein said replicating step comprises:

replicating an element 0 of the third register from an element 0 of the first register;

replicating, for all bits of an element 1 of the third register, a sign bit of the element 0 of the first register;

replicating an element 2 of the third register from an element 1 of the first register;

replicating, for all bits of an element 3 of the third register, a sign bit of the element 1 of the first register;

replicating an element 4 of the third register from an element 2 of the first register;

replicating, for all bits of an element 5 of the third register, a sign bit of the element 2 of the first register;

replicating an element 6 of the third register from an element 3 of the first register; and

replicating, for all bits of an element 7 of the third register, a sign bit of the element 3 of the first register.